DTCP Volume 1

Supplement E

Mapping DTCP to IP

(Informational Version)

Hitachi, Ltd.
Intel Corporation
Matsushita Electric Industrial Co., Ltd.
Sony Corporation
Toshiba Corporation

Revision 1.1
February 28, 2005

Preface

Notice

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Hitachi, Intel, MEI, Sony, and Toshiba (collectively, the “5C”) disclaim all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

Some portions of this document, identified as "Draft" are in an intermediate draft form and are subject to change without notice. Adopters and other users of this Specification are cautioned these portions are preliminary, and that products based on it may not be interoperable with the final version or subsequent versions thereof.

Intellectual Property

Implementation of this specification requires a license from the Digital Transmission Licensing Administrator.

Contact Information

Feedback on this specification should be addressed to dtla-comment@dtcp.com.

The Digital Transmission Licensing Administrator can be contacted at dtla-manager@dtcp.com.

The URL for the Digital Transmission Licensing Administrator web site is: http://www.dtcp.com.

Table of Contents

Figures

Tables

Volume 1 Supplement E DTCP Mapping to IP

V1SE.1 Introduction

This supplement describes the mapping of DTCP onto Internet Protocol (IP). All aspects of IEEE 1394 DTCP functionally are preserved except those described in Appendix D of Volume 1 which does not apply to this mapping and this supplement only details DTCP-IP specific changes or additions.

V1SE.1.1 Related Documents

This specification shall be used in conjunction with the following publications. When the publications are superceded by an approved revision, the revision shall apply.

- Digital Transmission Content Protection Specification Volume 1 and Volume 2
- RFC768 User Datagram Protocol
- RFC791 Internet Protocol
- RFC793 Transmission Control Protocol
- RFC1945 Hypertext Transfer Protocol – HTTP/1.0
- RFC2616 Hypertext Transfer Protocol – HTTP/1.1
- RFC1889 RTP: A Transport Protocol for Real-Time Applications

V1SE.1.2 Terms and Abbreviations

DTCP-IP	DTCP volume 1 Supplement E
DTCP Socket	Means the Socket used for AKE commands
E-EMI	Extended Encryption Mode Indicator
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
PCP	Protected Content Packet
RTP	Real-time Transport Protocol
RTT	Round Trip Time
Socket	Means IP-address concatenated with port number [e.g. <host>:<port>]
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

V1SE.2 Modifications to 4.2.3.2 Extended Format Fields (Optional Components of the Device Certificate)

For IP, the optional content channel cipher for AES-128 is not used.

V1SE.3 Modifications to Chapter 5 Restricted Authentication

Restricted authentication is not permitted for DTCP-IP transports.

V1SE.4 Modifications to Chapter 6 Content Channel Management Protection

V1SE.4.1 Modifications to 6.2.1 Exchange Keys

DTCP-IP requires only a single exchange key for all defined E-EMI.

V1SE.4.2 Modifications to 6.2.2.2 K_C for AES-128

The Content Key (K_C) is used as the key for the content encryption engine. K_C is computed from the three values shown below:

- Exchange Key K_X where only a single exchange key is used for all E-EMIs to protect the content.
- A random number N_C generated by the source device using RNG_F which is sent in plain text to all sink devices.
- Constant value C_{A0}, C_{B1}, C_{B0}, C_{C1}, C_{C0}, or C_{D0} which corresponds to an E-EMI value in the packet header.

The Content Key is generated as follows:

$K_C = \text{J-AES}(K_X, f[\text{E-EMI}], N_C)$ Where:

$f[\text{E-EMI}]$ {
$f[\text{E-EMI}] = C_{A0}$ when E-EMI = Mode A0
$f[\text{E-EMI}] = C_{B1}$ when E-EMI = Mode B1
$f[\text{E-EMI}] = C_{B0}$ when E-EMI = Mode B0
$f[\text{E-EMI}] = C_{C1}$ when E-EMI = Mode C1
$f[\text{E-EMI}] = C_{C0}$ when E-EMI = Mode C0
$f[\text{E-EMI}] = C_{D0}$ when E-EMI = Mode D0
}

C_{A0}, C_{B1}, C_{B0}, C_{C1}, C_{C0}, and C_{D0} are universal secret constants assigned by the DTLA. The values for these constants are specified in DTCP Specification available under license from DTLA.

Additional rules for AES-128 Cipher are described in the DTCP Specification available under license from the DTLA.

V1SE.4.2.1 Modifications to 6.2.2.2.1 AES-128 Related Key and Constant Sizes

Followings are the lengths of the keys and constants described above:

Key or Constant	Size (bits)
Exchange Key (K_X)	96
Scrambled Exchange Key (K_{SX})	96
Constants (C_{A0}, C_{B1}, C_{B0}, C_{C1}, C_{C0}, C_{D0})	96
Content Key for AES-128 Baseline Cipher (K_C)	128
Nonce for Content Channel (N_C)	64

Table 1 Length of Keys and Constants (Content Channel Management)

V1SE.4.3 Modifications to 6.3.1 Establishing Exchange Keys

It is mandatory that source devices expire an Exchange Key within 2 hours after all content transmission using PCP(s) has ceased.

It is mandatory that sink devices expire an Exchange Key within 2 hours of continuous non-use of that Exchange Key for decryption.

Source and sink devices must expire their Exchange Keys when they detect themselves being disconnected from all mediums. For wireless mediums this means when device detects that it is not connected to an access point or it is not directly connected to another device.

Source devices can not change or expire Exchange key during content transmission using PCP(s).

V1SE.4.4 Modifications to 6.3.2 Establishing Content Keys

For RTP transfers, source device generates a 64 bit random number as an initial value for N_C. N_C is updated periodically by incrementing it by 1 mod 2^{64} while at least on RTP transmission with PCP is in progress regardless of the value of E-EMI. The same value of N_C shall be used for all RTP simultaneous transmissions. The minimum period for update of the N_C is defined as 30 seconds, and the maximum period is defined as 120 seconds.

For HTTP transfers, source devices generate a 64 bit random number as an initial value of N_C for the initial TCP connection. The initial N_C for subsequent TCP connections must be different (another random number may be generated). If a HTTP response has more than 128 MB of content, N_C shall be updated every 128MB. N_C is updated by incrementing it by 1 mod 2^{64}. When plural HTTP responses are transmitted using the same TCP connection, N_C for subsequent HTTP response shall be updated from the latest N_C for the TCP connection.

V1SE.4.5 Modifications to 6.3.3 Odd/Even Bit

The Odd/Even Bit is not used in DTCP-IP as N_C value is sent with each PCP.

V1SE.4.6 Modifications to 6.4.1 Embedded CCI

Embedded CCI is carried as part of the content stream. Many content formats including MPEG have fields allocated for carrying the CCI associated with the stream. The definition and format of CCI is specific to each content format. Information used to recognize the content format should be embedded within the content.

V1SE.4.7 Modifications to 6.4.2 Encryption Mode Indicator (EMI)

E-EMI Mode	E-EMI Value	Description
Mode A0	1100_2	Copy-never (CN)
Mode B1	1010_2	Copy-one-generation (COG) [Format-cognizant recording only]
Mode B0	1000_2	Copy-one-generation [Format-non-cognizant recording permitted]
Mode C1	0110_2	Move (Audiovisual)
Mode C0	0100_2	No-more-copies (NMC)
Mode D0	0010_2	Copy-free with EPN asserted (CF/EPN)
N.A.	0000_2	Copy-free (CF)
	$----_2$	All other values reserved

Table 2 E-EMI Mode and E-EMI Description

V1SE.4.8 Modifications to 6.4.3 Relationship between Embedded CCI and EMI

E-EMI	Embedded CCI					
	CF	**CF/EPN**	**NMC**	**COG-AV**	COG-Audio	**CN**
Mode A0 (CN)	Allowed	Allowed	Allowed[1]	Allowed	Allowed	Allowed
Mode B1 (Format cognizant only recordable)	Allowed	Allowed	Prohibited	Allowed	Allowed	Prohibited
Mode B0 (Format non-cognizant recordable)	Allowed	Allowed	Prohibited	Allowed	Prohibited	Prohibited
Mode C1 (MOVE)	Prohibited	Prohibited	Allowed	Prohibited	Prohibited	Prohibited
Mode C0 (NMC)	Allowed	Allowed	Allowed	Allowed	Allowed	Prohibited
Mode D0 (CF/EPN)	Allowed	Allowed	Prohibited	Prohibited	Prohibited	Prohibited
N.A.	Allowed	Prohibited	Prohibited	Prohibited	Prohibited	Prohibited

Table 3 Relationship between E-EMI and Embedded CCI

[1] Not typically used.

V1SE.4.9 Modification to 6.4.4.1 Format-cognizant source function

Embedded CCI of programs					E-EMI
CF	CF/EPN	NMC	COG-AV	CN	
Don't care	Don't care	*[2]	Don't care	Present	**Mode A0**
Don't care	Don't care	Cannot be present	Present	Cannot be present	**Mode B1**
Don't care	Don't care	Cannot be present	Present	Cannot be present	**Mode B0**
Don't care	Don't care	Present	Cannot be present	Cannot be present	**Mode C0**
Don't care	Present	Cannot be present[3]	Cannot be present	Cannot be present	**Mode D0**
Present	Cannot be present	Cannot be present	Cannot be present	Cannot be present	**N.A.**
Other combinations					**Transmission Prohibited**

Table 4 Format-Cognizant Source Function CCI handling

V1SE.4.10 Modification to 6.4.4.2 Format-non-cognizant source function

E-EMI or recorded CCI[4] of source content	E-EMI used for transmission
Copy Never	Mode A0
COG: Format cognizant only recordable	Mode B1
COG: Format non-cognizant recordable	Mode B0
No-more-copies	Mode C0
EPN asserted Copy Free	Mode D0
Copy-Free	N.A.

Table 5 Format-Non-Cognizant Source Function CCI handling

V1SE.4.11 Modifications to 6.4.4.3 Format-cognizant recording function

E-EMI	Embedded CCI for each program				
	CF	CF/EPN	NMC	COG-AV	CN
Mode A0	Recordable	Recordable	Do not record	*[5]	Do not record
Mode B1	Recordable	Recordable	Discard entire content stream[6]	*[5]	Discard entire content stream[6]
Mode B0	Recordable	Recordable	Discard entire content stream[6]	*[5]	Discard entire content stream[6]
Mode C0	Recordable	Recordable	Do not record	Do not record	Discard entire content stream[6]
Mode D0	Recordable	Recordable	Discard entire content stream[6]	Discard entire content stream[6]	Discard entire content stream[6]

Table 6 Format-cognizant recording function CCI handling

[2] Don't care, but not typically used.

[3] This combination is allowed for format-non-cognizant source function, but is not permitted for format-cognizant source function.

[4] Recorded CCI is copy control information that is not embedded in the content program and does not require knowledge of the content format to extract.

[5] If the recording function supports recording a CCI value of No-more-copies then the CCI value of No-more-copies shall be recorded with the program. Otherwise the CCI of Copy-never shall be recorded with the program.

[6] If the function detects this CCI combination among the programs it is recording, the entire content stream is discarded.

V1SE.4.12 Modifications to 6.4.4.4 Format-cognizant sink function

E-EMI	Embedded CCI for each program				
	CF	CF/EPN	NMC	COG-AV	CN
Mode A0	Available for processing	Available for processing	Available for processing[1]	Available for processing	Available for processing
Mode B1	Available for processing	Available for processing	Discard entire content stream[7]	Available for processing	Discard entire content stream[7]
Mode B0	Available for processing	Available for processing	Discard entire content stream[7]	Available for processing	Discard entire content stream[7]
Mode C0	Available for processing	Available for processing	Available for processing	Available for processing[8]	Discard entire content stream[7]
Mode D0	Available for processing	Available for processing	Discard entire content stream[7]	Discard entire content stream[7]	Discard entire content stream[7]

Table 7 Format-cognizant sink function CCI handling

V1SE.4.13 Modification to 6.4.4.5 Format-non-cognizant recording function

E-EMI of the received stream	Recorded CCI[9] to be written onto user recordable media
Mode A0	Stream cannot be recorded
Mode B1	Stream cannot be recorded
Mode B0	No-more-copies
Mode C0	Stream cannot be recorded
Mode D0	EPN asserted Copy Free

Table 8 Format-non-cognizant recording function CCI handling

V1SE.4.14 Modification to 6.4.4.6 Format-non-cognizant sink function

Only bridge and rendering functions are allowed for this function unless the sink function is capable of processing the DTCP_descriptor.

V1SE.4.15 Modifications to 6.4.5.1 Embedded CCI for audio transmission

Value and Abbreviation	Meaning
11	Not defined
10 (COG-audio)	Copy-permitted-per-type
01 (NMC)	No-more-copies
00 (CF)	Copy-free

Table 9 Audio Embedded CCI Values

[7] If the function detects this CCI combination among the programs, the entire content stream is discarded.

[8] If the device has a rule for handling No-more-copies, this program shall be handled according to the rule. Otherwise the program shall be handled as Copy Never.

[9] Recorded CCI is copy control information that is not embedded in the content program and does not require knowledge of the content format to extract.

V1SE.4.16 Modifications to 6.4.5.3 Audio-format-cognizant source function

Embedded CCI of programs			E-EMI
CF	NMC	COG-audio	
Type specific[10]			**Mode A0**
Don't care	Cannot be present	Present	**Mode B1**
Don't care	Present	Don't care	**Mode C0**
Present	Cannot be present	Cannot be present	**N.A.**

Table 10 Audio-format cognizant source function CCI handling

V1SE.4.17 Modifications to 6.4.5.5 Audio-format-cognizant recording function

E-EMI	Embedded CCI of Program		
	CF	NMC	COG-audio
Mode A0	Recordable	Do not record	Recordable[11]
Mode B1	Recordable	Discard entire content stream[12]	Recordable[11]
Mode C0	Recordable	Do not record	Recordable[11]

Table 11 Audio-format-cognizant recording function CCI handling

V1SE.4.18 Modifications to 6.4.5.6 Audio-format cognizant sink function

E-EMI	Embedded CCI of program		
	CF	NMC	COG-audio
Mode A0	Available for processing	Available for processing	Available for processing
Mode B1	Available for processing	Discard entire content stream[12]	Available for processing
Mode C0	Available for processing	Available for processing	Available for processing

Table 12 Audio-format-cognizant sink function CCI handling

V1SE.4.19 Modifications to 6.4.5.8 Audio-Format-non-cognizant sink function

Only bridge and rendering functions are allowed for this function unless the sink function is capable of processing the DTCP_audio_descriptor.

V1SE.4.20 Modifications to 6.6.1 Baseline Cipher

For IP, the baseline cipher is AES-128 using the Cipher Block Chaining (CBC). AES-128 is described in FIPS 197 dated November 26, 2001 and the CBC mode is described in NIST SP 800-38A 2001 Edition.

---

[10] Usage is specified for each Audio type in Appendix A.

[11] The CCI value of No-more-copies shall be recorded with the program. Additional rules for recording are specified by each audio application in Appendix A.

[12] If the function detects this CCI combination among the programs it is recording the entire content stream is discarded.

V1SE.4.21 Modifications to 6.6.2.1 AES-128 Cipher

For AES-128, Cipher Block Chaining (CBC) is used. AES-128 is described in FIPS 197 dated November 26, 2001 and the CBC mode is described in NIST SP800-38A 2001 Edition. Additional rules for AES-128 Cipher are described in the DTCP specification available under license from DTLA.

V1SE.4.22 Modification to 6.6.3 Content Encryption Formats

DTCP encrypted content is sent via Protected Content Packets (PCP) where the format of the PCP is described in the following figure.

	msb							lsb
Header[0]	reserved (zero)			C_A	E-EMI			
Header[1]	exchange_key_label							
Header[2]	N_C (64 bits)							
Header[3]								
Header[4]								
Header[5]								
Header[6]								
Header[7]								
Header[8]								
Header[9]								
Header[10]	Byte length of content denoted as CL (32 bits)							
Header[11]								
Header[12]								
Header[13]								
EC[0]	Content affixed with 0 to 15 bytes of padding							
EC[1]								
EC[2]								
-								
-								
-								
EC[N-1]								

Figure 1 Protected Content Packet Format

Header [0]: C_A means cipher_algorithm where a value of 0_2 denotes AES-128 and the value 1_2 denotes optional cipher. **E-EMI** is as defined in section V1SE.4.7

Header [1]: Contains exchange_key_label which is described in the DTCP Specification available under license from DTLA.

Header [2..9]: Contains N_C as described in section V1SE.4.2.1.

Header [10..13]: Denotes byte length of content and does not include any padding bytes, where CL is less than or equal to 128 MB.

EC [0..N-1]: Represents encrypted frame and there is no EC when CL is zero otherwise it is a multiple of 16 Bytes in length where N = (Int((CL-1)/16)+1)*16 where padding length is equal to N-CL and Int(X) means maximum integer less than or equal to X. The value of each padding Byte is 00_{16}.

For RTP transfers, each RTP payload is encapsulated by a single PCP.

For HTTP transfers, responses may contain 1 or more PCPs.

V1SE.4.23 Modifications to 6.7.1 Move Function

This supplement defines a Move function in additional to the one described in section 6.7.1 where content with Embedded CCI of No-more-copies content may not be remarked as Copy-one-generation but instead be transmitted as No-more-copies using Mode C1 of E-EMI for IP transport of DTCP protected content and Recording functions may record the received content without remarking embedded CCI. E-EMI Mode B1 shall be used for Move-mode when source function uses Move function described in section 6.7.1. For clarity, the move function shall be used between a single source and a single sink function.

V1SE.5 Modifications to Chapter 8 (AV/C Digital Interface Command Set Extensions)

V1SE.5.1 Modifications to 8.1 Introduction

DTCP-IP uses TCP port to send/receive DTCP control packets, status command packets, and response packets. DTCP Socket identification of source device is described in section V1SE.10.2.

Devices shall wait at least one second for a response to a command before timing out.

V1SE.5.2 Modifications to 8.3.1 AKE Control Command

This section maps the AKE control command specified in Section 8.3.1 to the DTCP-IP Control Packet Format. Except as otherwise noted, the AKE control command sub fields used with IP have the same values and functions as detailed in Chapter 8.

	msb							lsb
Type[0]	0	0	0	0	0	0	0	1
Length[0]	(msb) Byte Length of Control and AKE_Info Fields (N+8)							
Length[1]								(lsb)
Control[0]	reserved (zero)				ctype/response			
Control[1]	Category = 0000_2 (AKE)				AKE_ID = 0000_2			
Control[2]	subfunction							
Control[3]	AKE_procedure							
Control[4]	exchange_key							
Control[5]	subfunction_dependent							
Control[6]	AKE_label							
Control[7]	number(option)				status			
AKE_Info[0..N-1]	AKE_Info							

Figure 2 DTCP-IP Control Packet Format

- Type, Length, and Control byte 0 are used to map DTCP to IP. Where the type field identifies version 1 AKE control packet.
- Ctype/response has the same values as referenced in chapter 8 of DTCP specification and specified by the AV/C Digital Interface Command.
- Control bytes 1..7 are identical to operand bytes 0..6 as specified in section 8.3.1, except for four most significant bits of Control byte 7 which is not used in IP.
- The AKE_Info field is identical to the data field specified in section 8.3.1.
- The AKE_label and source Socket of each control command should be checked to ensure that it is from the appropriate controller.

V1SE.5.3 Modification to 8.3.2 AKE Status Command

This section maps the AKE status command specified in Section 8.3.2 to the DTCP-IP Status Packet Format. Except as otherwise noted, the AKE status command sub fields used with IP have the same values and functions as detailed in Chapter 8.

	msb							lsb
Type[0]	0	0	0	0	0	0	0	1
Length[0]	(msb)	Byte length of Control						
Length[1]								(lsb)
Control[0]	reserved (Zero)				ctype/response			
Control[1]	category = 0000_2 (AKE)				AKE_ID = 0000_2			
Control[2]	subfunction							
Control[3]	AKE_procedure							
Control[4]	exchange_key							
Control[5]	subfunction_dependent							
Control[6]	AKE_label = FF_{16}							
Control[7]	number = F_{16}				status			

Figure 3 Status Packet Format

- Type, Length, and Control byte 0 are used to map DTCP to IP. Where the type field identifies version 1 AKE control packet.
- Ctype has the same values as referenced in Chapter 8 of DTCP specification and specified by the AV/C Digital Interface Command Set.
- Control bytes 1..7 are identical to operand bytes 0..6 as specified in Section 8.3.2.

V1SE.5.3.1 Modifications to AKE status command status field

Value	Status	Response code
0000_2	No error	STABLE
0001_2	Support for no more authentication procedures is currently available	STABLE
0111_2	Any other error	STABLE
1111_2	No information[13]	REJECTED

Table 13 AKE Status Command Status Field

[13] It is recommended that implementers not use the "No information" response.

V1SE.5.4 Modifications to 8.3.3

V1SE.5.4.1 AKE_ID dependent field

DTCP-IP implementations only require a single exchange key, specifically Bit 3 of exchange_key field will be used for transporting all DTCP Protected content over IP for all defined E-EMI.

For DTCP-IP both Source and Sink shall support only Full Authentication.

Therefore Restricted Authentication procedure (rest_auth) and Enhanced Restricted Authentication procedure (en_rest_auth) are prohibited. Extended Full Authentication procedure (ex_full_auth) is optional[14] and not used to handle Bit 3 of Exchange_key field.

Bit	AKE_procedure
0 (lsb)	Prohibited
1	Prohibited
2	Full Authentication procedure (full_auth)
3	Extended Full Authentication procedure[15] (ex_full_auth, optional)[16]
4 – 7 (msb)	Reserved for future extension and shall be zero

Table 14 AKE_procedure values

V1SE.5.4.2 Modifications to Authentication selection

	Sink supported authentication procedures	
Source supported authentication Procedures	**Full_auth**	**Full_auth and Ex_full_auth**
Full_auth	Full Authentication	Full Authentication
Full_auth and Ex_full_auth	Full Authentication	Extended Full Authentication

Table 15 Authentication selection

V1SE.5.4.3 Modification to Exchange_key values

DTCP-IP uses a single exchange key.

Bit	**Exchange_key**
0 (lsb)	Prohibited
1	Prohibited
2	Prohibited
3	Exchange key for AES-128
4 – 7 (msb)	Reserved for future extension and shall be zero

Table 16 Exchange_key values

[14] Features of this specification that are labeled as "optional" describe capabilities whose usage has not yet been established by DTLA.

[15] Devices that support extended device certificates use the Extended Full Authentication procedure described in this chapter.

[16] Features of this specification that are labeled as "optional" describe capabilities whose usage has not yet been established by the 5C.

V1SE.5.5 Modifications to AKE Subfunctions

Subfunction modified for DTCP-IP are described in the DTCP specification available under license from the DTLA.

V1SE.5.6 Modifications to 8.4 Bus Reset Behavior

If TCP connection is broken during authentication procedure, both source and sink devices shall immediately stop authentication procedure.

V1SE.6 Modifications to Appendix A (Additional Rules for Audio Applications)

V1SE.6.1 Modification to A.1 AM824 audio

Rules described in sections A.1.1, A.1.2, and A.1.2.3 are not limited to AM824 and Mode A is regarded as Mode A0 for DTCP-IP.

V1SE.6.1.1 Modification to A.1.1 Type 1: IEC 60958 Conformant Audio

Any content format with ASE-CCI equivalent to SCMS shall be regarded as Type 1 Audio.

V1SE.6.1.2 Modification to A.1.2 Type 2: DVD-Audio

Any content format containing DVD-Audio content and having ASE-CCI as described in Section A.1.2.2 shall be regarded as Type 2 Audio.

V1SE.6.1.3 Modification to A.1.3 Type 3: Super Audio CD

Any content format containing Super Audio CD content and having ASE-CCI equivalent to that described in Section A.1.3.2 shall be regarded as Type 3 Audio.

V1SE.6.2 Modification to A.2 MPEG Audio

Audio transmission via MPEG transport stream is permitted. Note that MPEG audio with ASE-CCI equivalent to SCMS is also Type 1 audio.

V1SE.7 Modification to Appendix B (DTCP_Descriptor for MPEG Transport Stream

V1SE.7.1 Modification to B.1 DTCP_descriptor

As no standardized method for carrying Embedded CCI in the MPEG-TS is currently available, the DTLA has established the DTCP_descriptor and DTCP_audio_descriptor to provide a uniform data field to carry Embedded CCI in the MPEG-TS. When MPEG-TS format audiovisual content is protected by DTCP, the DTCP_descriptor shall be used to deliver Embedded CCI information to sink devices. DTCP_audio_descriptor is defined for audio transmission which uses Type 1 Audio specified in Section V1SE.6.1.1.

V1SE.7.2 Modification to B.2 DTCP_descriptor syntax

DTCP_audio_descriptor is defined for audio transmission in addition to DTCP_descriptor defined in Section B.2. The first bit value of Private_data_type is used to distinguish DTCP_descriptor and DTCP_audio_descriptor.

In case of audio transmission, the following syntax is used, and DTCP_descriptor is referred to as DTCP_audio_descriptor.

The DTCP_audio_descriptor has the same syntax as DTCP_descriptor except for private_data_byte field. The definition of the private_data_byte field of the DTCP_audio_descriptor is as follows:

Syntax	Size(bits)	Formats
Private_data_type{		
Descriptor_ID	1	bslbf
Reserved	5	bslbf
DTCP_CCI_audio	2	bslbf
Audio_Type	3	bslbf
Reserved	5	bslbf
}		

Table 17 Syntax of private_data_type for DTCP_audio_descriptor

V1SE.7.2.1 Modification to B.2.1 private_data_byte Definitions:

Definition for the following fields is added for DTCP_audio_descriptor.

Descriptor_ID

This field indicates the kinds of descriptor.

Descriptor_ID	Meaning
0_2	DTCP_audio_descriptor
1_2	DTCP_descriptor

Table 18 Descriptor_ID

DTCP_CCI_audio

This field indicates the embedded CCI states for the transmission of Type 1 audio content.

DTCP_CCI_audio	Meaning
00_2	Copy-free
01_2	No-more-copies
10_2	Copy-permitted-per-type
11_2	Not defined

Table 19 DTCP_CCI_audio

Audio_type

This field indicates the Audio type.

Audio_type	Meaning
000_2	Type 1
$001_2..111_2$	Reserved for future extension

Table 20 Audio_type

V1SE.7.3 Modification to B.3 Rules for the Usage of the DTCP_descriptor

V1SE.7.3.1 Modification to B.3.1 Transmission of a partial MPEG-TS

For the audio transmission following rules are applied.

When a partial MPEG-TS that includes one or more programs is transmitted using DTCP, Audio-Format-cognizant source function shall insert the DTCP_audio_descriptor into the PMT[17] of each program for which ASE-CCI of Type 1 Audio is used and the ASE-CCI is not Copy-free. When the DTCP_audio_descriptor is inserted, it shall only be applied to the PMT.

An Audio-Format-cognizant source function shall set the DTCP_CCI_audio bits according to the ASE-CCI of Type 1 Audio provided for each program within the MPEG-TS. The DTCP_audio_descriptor shall be inserted into the program_info loop of the relevant PMT.

Additionally, if any of the Elementary Streams within a program are assigned specific ASE-CCI values of Type 1 Audio, Audio-format-cognizant source function shall set the DTCP_CCI_audio bits according to ASE-CCI of Type 1 Audio. The DTCP_audio_descriptor shall be inserted into the ES_info loop of the relevant PMT for the Elementary Stream.

[17] as described in the definition of ISO/IEC 13818-1

When Audio related content that is required to be treated as audiovisual content is transmitted as a part of Audio program, Audio-Format-cognizant source function, according to the upstream license, may insert DTCP_descriptor of the audio related contents to related ES_info loop in the Audio program.

V1SE.7.3.2 Modification to B.3.3.Treatment of the DTCP_descriptor by the sink device

This section replaces Section B.3.3 and describes the treatment of the DTCP_descriptor and DTCP_audio_descriptor when received by a sink device. When the function of the sink device is format cognizant and receives recognizable Embedded CCI other than the DTCP_descriptor and DTCP_audio_descriptor within an MPEG-TS, the alternative Embedded CCI shall take precedence over the information contained within the DTCP_descriptor or DTCP_audio_descriptor. Furthermore, the DTCP_descriptor and DTCP_audio_descriptor are only valid when they are inserted into the PMT. If a DTCP_descriptor or DTCP_audio_descriptor is found in another location, it shall be ignored.

When the only Embedded CCI detected is the DTCP_descriptor or DTCP_audio_descriptor, the DTCP_descriptor shall be regarded as the Embedded CCI described in Sections V1SE.4.11 and V1SE.4.12 except as otherwise noted, and the DTCP_audio_descriptor shall be regarded as the Embedded CCI described in Sections V1SE.4.18 , and interpreted as follows:

- If a DTCP_descriptor or DTCP_audio_descriptor is found in an ES_info loop of the PMT, the Embedded CCI value contained in the descriptor should only be used as the CCI for the specific ES for which the DTCP_descriptor or DTCP_audio_descriptor is associated.
- When the only Embedded CCI detected in an ES_info loop of an Audio program is DTCP_descriptor, the DTCP_descriptor shall be regarded as the Embedded CCI described in only Section V1SE.4.12.
- If a DTCP_descriptor and DTCP_audio_descriptor is not found in the ES_info loop for a specific ES, but is instead found in the program_info loop, the Embedded CCI values contained within the DTCP_descriptor or DTCP_audio_descriptor shall be used as the CCI for that ES.
- A program in a stream shall be regarded as Copy-free if the stream contains multiple programs and none of Embedded CCI, DTCP_descriptor and DTCP_audio_descriptor is detected in the program and a DTCP_descriptor or DTCP_audio_descriptor is detected in another program on the same stream.

V1SE.8 Additional Requirements

V1SE.8.1 Authentication Capability Constraint

For DTCP-IP both source and sink devices shall only use Full Authentication.

V1SE.8.2 Internet Datagram Header Time To Live (TTL) Constraint

TTL is described in RFC791 and the following requirements only apply to IP datagrams that transport DTCP AKE commands. Transmitting devices shall set TTL value of such transmitted IP datagrams to a value no greater than 3 and correspondingly receiving devices shall discard such received IP datagrams which have a TTL value greater than 3.

V1SE.8.3 802.11 Constraint

DTCP devices with integrated 802.11 must ensure that either WEP or other such equivalent protection mechanism (e.g. WPA or WPA2) is engaged prior to exchanging DTCP AKE commands and protected content via such an network interface. For interoperability purposes devices must have at least WEP capabilities. Please note that this requirement to use WEP may be amended to require use of successor technologies as designated by DTLA.

V1SE.8.4 DTCP-IP Move Protocol

Transaction-based is to be defined.

V1SE.8.5 Additional Localization via RTT

Source and sink devices must implement Additional Localization as specified in this section.

Source devices with Additional Localization (AL) when conducting an AKE with a Sink device with AL must perform a RTT test if the sink device's Device ID is not on the source device's RTT registry.

Source devices will add a Sink device's Device ID to the Source device's RTT registry, will set the content transmission counter for the sink device to 40 hours, and will provide an exchange key only if the source device measures a RTT value of 7 milliseconds or less during RTT test.

Source devices when transmitting content will update content transmission counters of all RTT registered sink devices and are required to remove the Device ID of a sink device from the RTT registry after counting 40 hours of content transmission.

Background RTT testing is not a required capability. If background RTT testing is supported, the source device will add the sink device's Device ID to the RTT registry if not registered and set content transmission counter to 40 hours only if the source device measures a RTT value of 7 milliseconds or less during RTT test.

V1SE.8.5.1 Protected RTT Protocol

DTCP-IP's protected RTT protocol is described in Figure 4 and is used in RTT-AKE and Background RTT check procedures. The RTT protocol is executed after the Challenge-Response portion of the AKE is completed. SHA-1 is used to construct following messages that are exchanged during RTT testing protocol to ensure that source and sink which completed Challenge-Response portion of AKE are only ones involved in RTT testing.

- $MAC1A = MAC1B = [SHA\text{-}1(MK+N)]_{msb80}$
- $MAC2A = MAC2B = [SHA\text{-}1(MK+N)]_{lsb80}$

- OKMSG = $[SHA\text{-}1(MK+N+1)]_{msb80}$
  Where MK is 160 bits and equal to SHA-1(Kauth||Kauth), N is 16 bit number that ranges from 0 to 1023, and "+" used in RTT Protocol means mod 2^{160} addition.

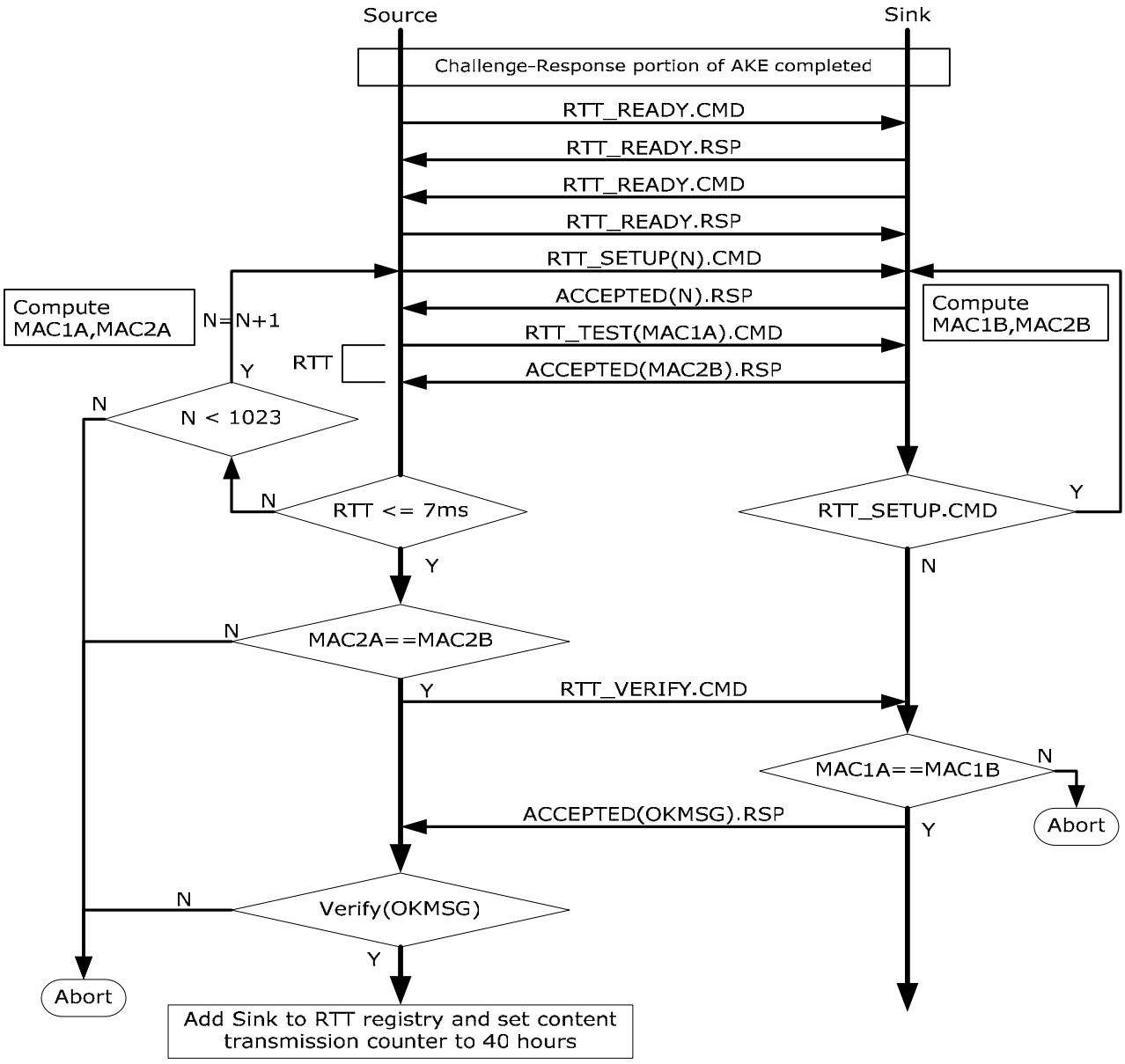

Figure 4 RTT Protocol Diagram

The RTT_READY command is used to indicate that authentication computation is complete and that source and sink devices are ready to execute the RTT test procedure.

The RTT procedure begins by first establishing value of N using the RTT_SETUP command. N is initially set to zero and can range from 0 to 1023 as maximum permitted RTT trials per AKE is 1024.

After preparation of MAC values corresponding to N, source device will then measure RTT which is the time interval starting after source transmits RTT_TEST command and terminates upon reception of RTT_TEST accepted response.

If the RTT is greater than 7 milliseconds and the value of N is less than 1023 the source will repeat RTT procedure by incrementing N by 1 and reissue RTT_SETUP and RTT_TEST commands.

If the measured RTT is less than or equal to 7 milliseconds:

The source device compares most recently computed MAC2A to most recently received MAC2B and if not equal the source device aborts RTT procedure else if equal it sends RTT_VERIFY command to sink device.

The sink device will after receipt of RTT_VERIFY command compare the most recently received MAC1A and most recently computed MAC1B and if not equal aborts RTT procedure else if equal it will send OKMSG in RTT_VERIFY accepted response.

The source device will verify OKMSG and if it is not correct the source device aborts RTT procedure else it will add sink device's Device ID to RTT registry and set content transmission counter to 40 hours.

If RTT procedure is aborted the source shall not provide an exchange key.

V1SE.8.5.2 RTT-AKE

The RTT-AKE procedure starts exactly the same as normal AKE but source and sink devices that have DTCP certificates with AL flag set to one must check AL flag value of other device and if the AL flag value is also set to one then:

The sink device after completing Challenge-Response portion of AKE will wait and the sink device will abort if it receives any other command than the RTT_READY command, EXCHANGE_KEY command, or AKE_CANCEL command.

The source device then examines the RTT registry and if the sink device's Device ID is on its RTT registry, the source device proceeds to exchange key portion of AKE otherwise the source device initiates a RTT test procedure and if during test it obtains a RTT measurement of 7 milliseconds or less it will add the sink device's Device ID to its RTT registry, set content transmission counter to 40 hours, and then proceed to exchange key portion of AKE.

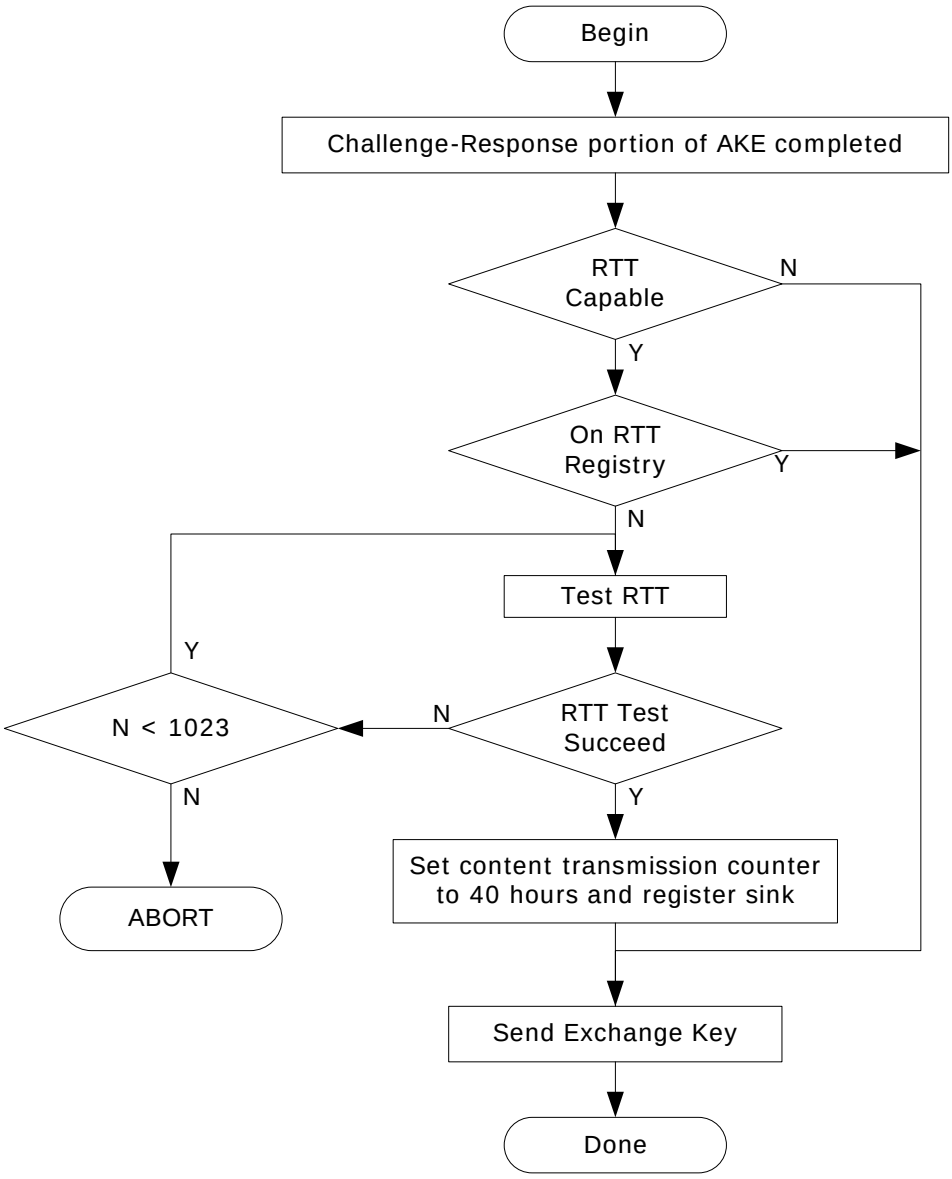

Figure 5 AKE-RTT Informative Flow Diagrams

V1SE.8.5.3 Background RTT Check

The Background RTT check procedure permits either the source or sink device to initiate an RTT background check which is only used to add sink device to source device's RTT registry if not on RTT registry or if already on the source device's RTT registry set the count transmission counter to 40 hours. In case of Background RTT check source devices shall not transmit an exchange key.

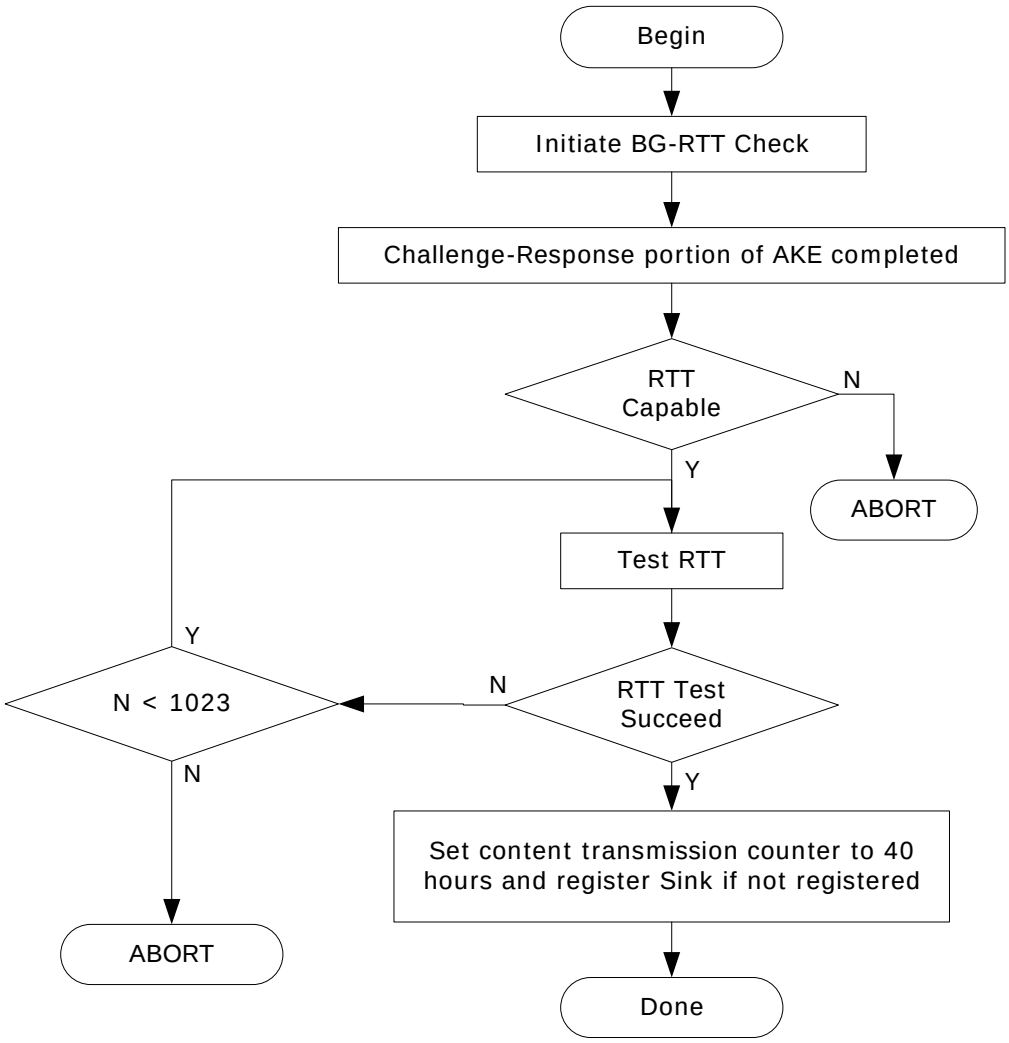

Figure 6 Background RTT Check Informative Flow Diagram

V1SE.8.6 Content Key Confirmation

For interoperability the content key confirmation function is limited to only those source and sink devices whose AL flag has a value of one. The sink device uses the CONT_KEY_CONF subfunction to confirm that the content key via the associated N_C is current.

Sink devices must monitor and confirm the N_C value of the most recently received PCP containing encrypted content for each content stream and then periodically reconfirm subsequent N_C(s) at least every 2 minutes. Periodic confirmation of N_C can be avoided if after initial confirmation the sink monitors and confirms that subsequent N_C values are monotonically increasing contiguous values.

Per content stream, sink devices after an initial non-confirmation of a N_C have one minute to repeatedly attempt to confirm a subsequent N_C value before they must terminate decryption for that content stream.

Sink devices may restart decryption upon confirmation of any N_C after a N_C non-confirmation event.

The content key confirmation procedure requires the sink device to send the N_C value under test (N_CT) to the source device. Upon receipt the source device checks the received N_CT against its current N_C values and if any are within the range N_CT to N_CT+5 then it confirms that N_CT is valid. The confirmation procedure is depicted in following figure.

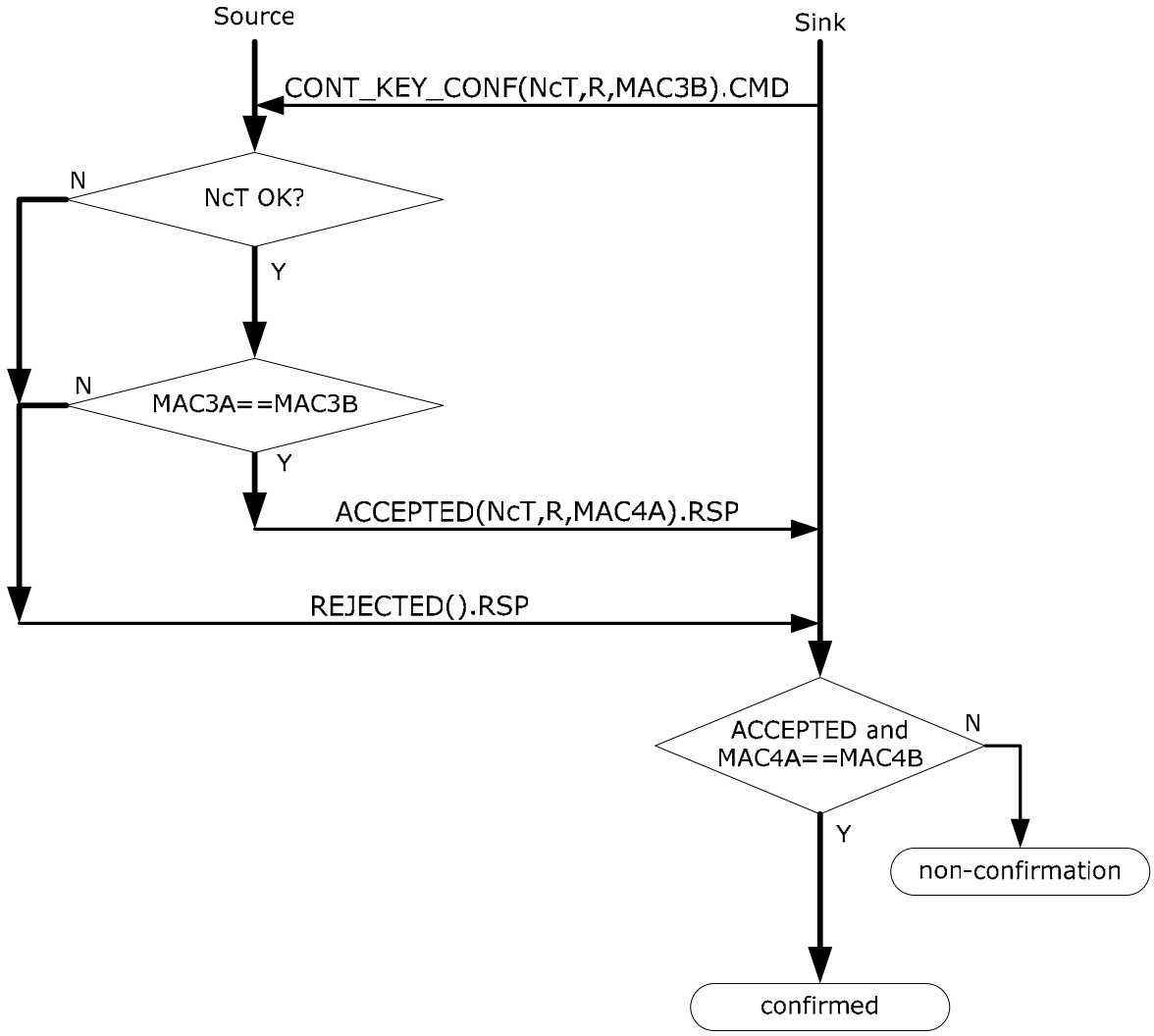

Figure 7 Content Key Confirmation Procedure

Where:

MX=SHA-1(Kx||Kx),

R is 64 bits, its initial value is a random number and is incremented by 1 mod 2^{64} for subsequent trials.

MAC3A = MAC3B = [SHA-1(MX+NcT+R)]msb80

MAC4A = MAC4B = [SHA-1(MX+NcT+R)]lsb80

"+" used in the above formulas means mod 2^{160} addition

V1SE.9 Additional Commands and Sequences

These additions defined for DTCP-IP are described in the DTCP specification available under license from the DTLA.

V1SE.10 Recommendations

V1SE.10.1 Recommended MIME type for DTCP protected content

DTCP application media type is as follows:

application/x-dtcp1; CONTENTFORMAT=<mimetype>

Where **CONTENTFORMAT**, is the standard content media type that is protected by DTCP.

In addition, information identifying DTCP Socket may be included as follows:

application/x-dtcp1; DTCP1HOST=<host>; DTCP1PORT=<port>; CONTENTFORMAT=<mimetype>

Refer to V1SE.10.2.1 for description of **DTCP1HOST** and **DTCP1PORT**.

Content type of HTTP response is set to DTCP application media type.

V1SE.10.2 Identification of DTCP Sockets

DTCP uses a TCP port to support various command and control protocols (i.e. AKE, Exchange Keys, SRM,,,) and either TCP or UDP for content transport. This section details recommend practices for identifying DTCP Sockets.

V1SE.10.2.1 URI Recommended Format

This following information is inserted into the query string portion of URI and is used to communicate the source's content and DTCP Socket to the sink. The source obtains the sink's DTCP Socket when the sink establishes a TCP connection to the source.

<service>://<host>:<port>/<path>/<FileName>.<FileExtention>?CONTENTPROTECTIONTYPE=DTCP1&DTCP1HOST=<host>&DTCP1PORT=<port>

Where:

CONTENTPROTECTIONTYPE, is set to "DTCP1" where 1 represents a DTCP-IP version number that can be incremented in future as the needed.

DTCP1HOST specifies the IP address and **DTCP1PORT** specifies the port number of the DTCP Socket of the source device.

V1SE.10.2.2 HTTP response

Content type of HTTP response is set to DTCP application media type as follows:

Content-Type: application/x-dtcp1 ; DTCP1HOST=<host> ; DTCP1PORT=<port> ; CONTENTFORMAT=<mimetype>

V1SE.10.3 Header Field Definition for HTTP

The following header fields are defined for HTTP transfers.

V1SE.10.3.1 Range.dtcp.com

The Range.dtcp.com header is used in the same manner as the RANGE header defined in RFC 2616 except that range specification applies to the content before DTCP processing.

V1SE.10.3.2 Content-Range.dtcp.com

The Content-Range.dtcp.com header is used in the same manner as the CONTENT-RANGE header defined in RFC 2616 except that range specification applies to the content before DTCP processing.